

Implantación de una asignatura sobre programación de FPGAs en un Máster de Ingeniería de Telecomunicación

Manuel Arias, Diego G. Lamar, Alberto Rodríguez, Aitor Vázquez, Ignacio Castro, María Rodríguez

Grupo de Sistemas Electrónicos de Alimentación

Edificio 3, Campus de Viesques s/n

Gijón, SPAIN

ariasmanuel@uniovi.es

Resumen— En este artículo se presenta la planificación de las clases teóricas, su contenido y el planteamiento de las prácticas de una asignatura de máster centrada en la programación de FPGAs (Field Programmable Gate Arrays) mediante lenguaje VHD (VHDL). El objetivo es obtener una asignatura perfectamente coordinada tanto con las asignaturas del grado como con las asignaturas del máster en el que se enmarca, siendo además otro objetivo la coherencia con el planteamiento general del máster.

Palabras clave—máster, asignatura presencial, FPGAs, telecomunicaciones

I. INTRODUCCIÓN

El desarrollo de una asignatura enmarcada dentro de los másteres nacidos bajo el Espacio Europeo de Educación Superior (EEES) tiene una serie de requisitos y condiciones fundamentales para tener éxito y ser de utilidad a los alumnos egresados y al tejido empresarial, el receptor último de dichos alumnos. Entre estos aspectos cabe citar un carácter eminentemente práctico, con asignaturas donde primen los casos prácticos y reales y el trabajo en el laboratorio [1]. El trabajo autónomo es otra de las actuales señas de identidad que debe tener un máster de ingeniería [2]-[3], con planteamientos donde las clases expositivas sean sólo el punto de partida del aprendizaje del alumno. Asimismo, es precisa una gran coordinación de contenidos entre las distintas asignaturas que conforman un determinado máster [4]. Con igual importancia, esta coordinación también debe buscarse con los grados desde los que se puede acceder a dicho máster, especialmente en el caso de aquellos denominados generalistas o con atribuciones. Esto último resulta complejo dada la incipiente movilidad de los alumnos entre distintas escuelas en el paso del grado al máster [5]. Dicho de otro modo, no es extraño que el centro en el que un alumno ha cursado el grado sea distinto del centro en el que estudia el máster. Esto implica que los másteres deberían coordinarse con grados de diversas universidades y especialidades, lo cual resulta imposible. No obstante, sí que es posible coordinar los másteres y sus asignaturas, al menos, con los grados de la misma universidad (o incluso universidades cercanas). Por último, otro aspecto clave que deben tener los másteres es que ellos y sus asignaturas deben ser acordes a las actuales demandas de las empresas receptoras de ingenieros y de la sociedad en general, tanto en las competencias

específicas, marcadas por las necesidades tecnológicas del momento, como en las transversales, más determinadas por los métodos de trabajo actuales dentro de las empresas [6]-[7].

En este artículo se presenta el planteamiento de la asignatura Diseño Digital Avanzado (DDA), que trata de cumplir con todas las condiciones antes mencionadas. Se trata de una asignatura centrada en la programación de *Field Programmable Gate Arrays* (FPGAs) mediante el lenguaje de descripción de *hardware* VHDL (del término sajón *VHSIC Hardware Description Language*).

El planteamiento de este artículo es el siguiente. En el apartado II se contextualiza el máster y la asignatura, describiendo las asignaturas que lo componen, el tipo de alumnos que se matriculan en el mismo, la duración de la asignatura, su reparto de horas lectivas, etc. Asimismo, se hace una descripción de los objetivos que se persiguen. En el apartado III se detalla la organización de temario y la metodología seguida en las clases de teoría con el fin de cumplir los objetivos planteados. También se detallan las prácticas de laboratorio y su coordinación con las clases expositivas. En el apartado IV se evalúan los resultados obtenidos en la asignatura y las conclusiones extraídas, incluyendo los resultados de los alumnos en su evaluación y los resultados obtenidos en una encuesta anónima sobre la asignatura. También se incluyen las posibles mejoras planteadas por los alumnos y las acciones futuras para el próximo curso académico.

II. CONTEXTUALIZACIÓN DEL MÁSTER Y LA ASIGNATURA

La asignatura en la que se centra este artículo, Diseño Digital Avanzado, se enmarca en el Máster Universitario en Ingeniería de Telecomunicación que se imparte en la Universidad de Oviedo. Se asocia a los ISCED Electrónica y Automática y Electricidad y Energía. Es importante destacar que se trata de un máster habilitante para la profesión regulada de Ingeniero de Telecomunicación. Los créditos del máster son 120, repartidos en 90 créditos obligatorios, 24 de trabajo fin de máster, 3 créditos optativos y 3 créditos de prácticas externas.

La duración del máster es de dos años y en la Figura 1 puede verse un esquema con la distribución de sus bloques o materias de conocimiento: sistemas de telecomunicación, sistemas electrónicos, ingeniería telemática, transmisión digital, integración de sistemas y tecnologías y dirección de

Semestre 1	Sist. Telecomunicación	Sist. Electrónicos	Ing. Telemática	Transmisión digital
Semestre 2				
Semestre 3	Integr. de sistemas y tecnologías	Dirección de proyectos		
Semestre 4	Optativas	Prácticas	Trabajo Fin de Máster	

Figura 1. Esquema de distribución de los bloques del Máster

Semestre 1	Diseño subsistemas de comunicaciones	Circuitos para comunicac.	Internet de nueva generación	Transmisión de datos
Semestre 2	planificación y despliegue	Diseño Digital Avanzado	Aplicaciones y sist. Distrib.	Comunicaciones adaptativas
Semestre 3	Integr. Sist. Electrónicos	Diseño Microelectrónico	Administr. de redes y serv.	
Semestre 4	Optativas	Prácticas	Trabajo Fin de Máster	

Figura 2. Esquema de distribución de las asignaturas del máster por semestre y bloque

proyectos. A esto se debe sumar el trabajo fin de máster, las prácticas, etc.

La asignatura DDA tiene carácter obligatorio, se enmarca en el primer cuatrimestre del primer año (véase Figura 2) y consta de 6 créditos ECTS. La distribución de trabajo presencial del alumno es de 28 horas de clases expositivas (2 horas a la semana) y 16 horas de prácticas de laboratorio (véase apartado IV en relación al número de horas de clases expositivas).

Durante los dos años de existencia de la asignatura DDA, el número de alumnos ha sido 12 el primer año y 13 el siguiente. Estos alumnos vienen de cualquiera de las tres especialidades que presenta el Grado en Ingeniería en Tecnologías y Servicios de Telecomunicación de la Universidad de Oviedo: Sistemas de Telecomunicación, Sistemas Electrónicos y Telemática. Asimismo, 6 de los alumnos provenían de otras Universidades y/o Grados.

Los objetivos perseguidos en la asignatura son claros:

- Que el alumno comprenda el funcionamiento, esquema interno, ventajas y desventajas de las FPGAs
- Que el alumno tenga unos conocimientos claros y extensos de VHDL y sea capaz de emplearlos en la programación de FPGAs para desarrollar cualquier programa, independientemente de sus especificaciones.
- Que sea capaz de adaptarse a cualquier tipo de FPGA y a cualquier entorno de diseño y programación, no sólo a los empleados en la asignatura.
- Que desarrolle la habilidad de aprendizaje autónomo, al menos, en el ámbito de la programación VHDL y las FPGAs

III. ORGANIZACIÓN DE LA ASIGNATURA Y RELACIÓN CON EL RESTO DE ASIGNATURAS DEL MÁSTER

En este apartado se detalla primeramente el punto de partida de la asignatura y su relación con las asignaturas de grado. A continuación se explica la organización de los contenidos teóricos. Posteriormente, el planteamiento que se

sigue con las prácticas de aula y las prácticas de laboratorio y la finalidad que se persigue con ellas de cara a la formación del alumno. Seguidamente se explica la metodología seguida para tratar de vincular y relacionar esta asignatura con los contenidos de las otras asignaturas del máster. Por último, se explica el método de evaluación de los alumnos.

A. Punto de partida

Como se ha comentado anteriormente, la coordinación de la asignatura con el resto de asignaturas del máster resulta fundamental a la hora de determinar sus contenidos y forma de plantear las clases y prácticas. No obstante, es igual de importante la coordinación con las asignaturas de los grados desde los que se puede acceder al mismo. Esto no es una tarea fácil, ya que actualmente los alumnos muestran bastante movilidad y eso implica una coordinación con los grados de otras universidades y no sólo con los propios. Por lo tanto, esta coordinación sólo puede lograrse mediante un análisis global, lo que implica que con ciertas universidades la coordinación puede no ser total. Los grados tienden a plantear la docencia desde una perspectiva gradual, coordinada y, en cierta medida, similar a como se hacía antes de la adaptación al EEES, lo cual facilita enormemente la tarea de coordinación. Típicamente, los alumnos de grado suelen comenzar con la electrónica analógica y continúan con la digital discreta (puertas lógicas, multiplexores, biestables, etc.). A continuación, en el mismo curso o en otro posterior, suelen centrarse en los microcontroladores y microprocesadores. Esta afirmación es aplicable a bastantes universidades ya que resulta ser la evolución lógica y natural, pues además de haber visto la electrónica digital discreta, los alumnos han cursado ya asignaturas de programación en lenguaje C o equivalente. Esto permite, en unos casos, centrarse en la explicación del microcontrolador y sus periféricos y, en otros casos, además de lo anterior, en la programación en lenguaje ensamblador. A partir de ahí, los alumnos de Ingeniería de Telecomunicación pasan a centrarse en la programación de Procesadores de Señal Digital o DSPs (evolución natural desde la programación de microcontroladores de gama media) y/o en la programación de CPLDs (Complex Programmable Logic Devices) o FPGAs. Esto último suele ser lo menos habitual, al menos de forma exclusiva (sin ver DSPs). Por lo tanto, puede suponerse que los alumnos que acceden al máster de ingeniería de telecomunicación apenas tienen conocimientos de VHDL o estos son limitados (en el caso de haber repartido el tiempo entre DSPs y FPGAs).

Por todo lo planteado anteriormente, los contenidos de la asignatura DDA parten de la explicación más básica (estructura interna de la FPGA) y avanzan hasta llegar a los aspectos más avanzados de la programación VHDL.

B. Organización de contenidos

La asignatura se centra en 6 grandes bloques:

- Introducción a las FPGAs y descripción básica del VHDL
- Elementos básicos y concepto de entidad y arquitectura
- Programación concurrente

- Programación secuencial
- Programación jerárquica
- Aspectos avanzados

En el primero de ellos, los alumnos adquieren un conocimiento de la estructura interna de las FPGAs (estructura de los elementos lógicos, relación entre ellos, organización de bloques, elementos adicionales internos como PLLs, memorias, multiplicadores, etc.). Este estudio se realiza sobre dos FPGAs, una de Xilinx y otra de Altera, con el objeto de dejarles claro que aunque en prácticas se usará solamente una de ellas, la estructura interna es muy similar. También se trata de dar una idea clara del VHDL. Dos aspectos fundamentales que se remarcan en este bloque son:

- La estructura jerárquica y basada en unidades de diseño del VHDL. Se trata de dar una visión muy general de cómo son los programas escritos en VHDL. La importancia de esta visión general se explica más adelante en el texto.
- El carácter *hardware* de los programas escritos en lenguaje VHDL. Los alumnos llegan a esta asignatura con una base muy extensa de programación de microcontroladores, microprocesadores, desarrollo de *scripts* para *Matlab*® u otros programas matemáticos, etc. Es decir, tiene una mente organizada para el desarrollo de programas *software* de ejecución secuencial. Resulta fundamental por tanto remarcarles que VHDL describe o “programa” *hardware*, lo que hace que efectos como retardos en líneas eléctricas pasen a tener relevancia. También deben ser conscientes de otro hecho fundamental: en un lenguaje de programación “tradicional”, la traducción de las instrucciones a lenguaje ensamblador (y código máquina) no supone ningún tipo de problema, siempre es posible dicha traducción. En cambio, una descripción realizada mediante VHDL puede que desempeñara la tarea deseada de forma correcta, pero el compilador puede no ser capaz de traducirla a *hardware*. Es decir, se debe remarcar que en VHDL no existe una correspondencia biunívoca entre una instrucción y una estructura *hardware* particular. Asimismo, también es fundamental que sean conscientes de que ciertas instrucciones escritas en VHDL tienen carácter concurrente y se ejecutan “a la vez”. Si esto no se remarca, tienden a programar en VHDL como si programaran en cualquier otro lenguaje (ensamblador, C, etc.), con todos los errores y problemas que ello implica.

En el siguiente bloque (elementos básicos y concepto de entidad y arquitectura), se instruye a los alumnos en los operadores, objetos (variables, señales y constantes) e instrucciones para la creación de tipos, subtipos, etc. Es decir, los elementos constructivos básicos del VHDL. A continuación, se detalla el concepto de entidad y arquitectura y todas las instrucciones asociadas a su creación (puertos, genéricos, orden de creación, etc.). El motivo de explicar los elementos de más bajo nivel y, a la vez, los de más alto se debe a uno de los objetivos buscados en esta asignatura: que el

alumno sea capaz de programar una FPGA y desarrollar cualquier proyecto basado en dicho lenguaje. Para cumplir dicho objetivo, el alumno debe practicar lo más posible. Con un planteamiento *top-to-bottom*, donde se comienza por las entidades de más alto nivel (paquetes, entidades y arquitecturas) y se desciende hasta los objetos básicos, los alumnos tienen una visión general del conjunto desde el comienzo, pero no pueden programar hasta que no llegan a los elementos básicos, que son la base de la programación. Por el contrario, con un planteamiento *bottom-to-top* los alumnos pueden desarrollar programas de una complejidad creciente a medida que avanzan las explicaciones, pero les faltaría la visión de conjunto hasta llegar al final de las mismas, donde verían el concepto de entidad y arquitectura. Con el planteamiento propuesto, los alumnos tienen la visión de conjunto necesaria pero pueden empezar a desarrollar programas desde el principio: en un comienzo, programas sencillos y, a medida que conocen más instrucciones y consejos de programación, ir desarrollando programas más complejos.

Los bloques relativos a las instrucciones concurrentes y secuenciales pueden considerarse el núcleo principal de la asignatura. El hecho de que las instrucciones concurrentes se expliquen antes de las secuenciales no es aleatorio. Como ya se ha dicho, los alumnos no son conscientes de la gran diferencia que supone el VHDL frente a los lenguajes de programación tradicionales a los que están acostumbrados (C, *scripts* de *MatLab*®, etc.). La explicación de estas instrucciones concurrentes, mucho más ligadas al carácter *hardware* de las FPGAs, permite aclarar a los alumnos las principales diferencias del VHDL con otros lenguajes: las instrucciones describen la configuración del *hardware* de la FPGA, lo que hace que dichas instrucciones se ejecuten de forma concurrente. Además, también permite aclarar que no siempre las instrucciones se “traducen” de la misma forma, a diferencia de lo que ocurre con los lenguajes de programación de *software*. Asimismo, el hecho de explicar primero las instrucciones concurrentes responde a una estrategia docente: que los alumnos usen dichas instrucciones. En caso contrario, los alumnos tienden a programar siempre empleando instrucciones secuenciales ya que son a las que más acostumbrados están. Aunque esto es una práctica aceptable e incluso recomendada por ciertos autores, termina por hacer que los alumnos no comprendan ni sepan usar las instrucciones concurrentes. De esta forma, los alumnos practican y conocen todas las instrucciones y luego pueden decidir qué estilo de programación quieren seguir.

El bloque de programación jerárquica explica las instrucciones relativas a la generación de componentes y su uso en otras arquitecturas. No obstante, la parte más relevante de este bloque es que los alumnos comprendan claramente el carácter jerárquico del VHDL y la capacidad de reaprovechamiento de código.

Por último, en el bloque de aspectos avanzados se detallan todos los elementos que no son fundamentales para el desarrollo de un código en VHDL pero que son interesantes y relevantes. Funciones y procedimientos, lógica multivaluada, alias, sobrecarga de operadores, máquinas de estado [8], etc. son algunos ejemplos de los aspectos recogidos en este bloque. Como se puede apreciar, no son fundamentales para poder

escribir programas en VHDL, pero sí resultan fundamentales si se quiere tener una visión completa del VHDL y aprovechar al máximo su potencial.

C. Planteamiento de las prácticas de aula

Como puede apreciarse en II, la asignatura no cuenta con horas para prácticas de aula. Esto carece de sentido y es por ello que de las 28 horas de clases expositivas, 8 se dedican a prácticas de aula, básicamente realización y corrección de ejercicios. El planteamiento de las prácticas de aula, en esta asignatura, no entraña ninguna complejidad y la aplicación directa del aprendizaje basado en problemas ofrece buenos resultados [9]-[12]. Sí es interesante el hecho de que se pueden diferenciar tres tipos de prácticas de aula: ejemplos, ejercicios cortos y ejercicios largos. Los ejemplos persiguen enfatizar o aclarar un aspecto puntual de la programación VHDL (por ejemplo, diferencia entre señal y variable y su comportamiento dispar dentro de los procesos). Los ejercicios cortos persiguen afianzar conocimientos generales y para su resolución los alumnos disponen de 5-10 minutos antes de que el profesor exponga y comente la solución (a diferencia de los ejemplos, en los que el profesor plantea una o dos preguntas muy puntuales y que son resueltas por los alumnos en 1-2 minutos). Los ejercicios largos siguen un planteamiento distinto. Al finalizar un tema, o en puntos determinados del mismo, el profesor plantea algunos ejercicios más largos y que requieren más tiempo. Dichos ejercicios deben ser resueltos por los alumnos en casa, donde pueden emplear la herramienta de programación para desarrollarlos. Pasados unos días, el profesor explica en clase la solución. Es importante remarcar que en este caso se intenta que muchos de los ejercicios tengan una clara aplicación real y, en la medida de lo posible, ésta guarde relación con el resto de asignaturas del máster. Asimismo, todos los ejercicios propuestos pueden programarse en la tarjeta de práctica (véase III.D) y comprobar su funcionamiento real.

No tiene sentido dar una descripción detallada de todos los ejercicios cortos y ejemplos planteados en la asignatura, pero sí dar un ejemplo de cada uno de ellos y los objetivos que persiguen:

- Ejercicio corto. La explicación de cada instrucción del lenguaje VHDL (*case*, *when*, *with-select*, *if*, etc.) incluye como ejercicio corto la implementación de un decodificador 3 a 8. Al margen de los objetivos obvios, el hecho de que tengan que programar siempre el mismo elemento les permite apreciar claramente las diferencias que existen en cada instrucción, sin verse influenciados por el elemento que implementan. Dicho de otro modo, pierden ese grado de libertad en pos de una mejor comprensión de las instrucciones. Lógicamente, a estos ejercicios cortos siguen otros (o ejemplos) en los que el elemento a programar es distinto y fuerza a los alumnos a pensar.
- Ejemplo. Se plantea a los alumnos dos programas que teóricamente desempeñan la misma función (cálculo de un bit de paridad). Deben encontrar el que no funciona y el motivo, que no es otro que el hecho de haber usado una señal en un proceso y haber desarrollado

éste pensando en que el valor de dicha señal se actualiza de forma inmediata (cuando realmente lo hace al salir del proceso). El tiempo que se deja para encontrar esta respuesta, y ese es el motivo de considerarlo ejemplo y no ejercicio, es de un minuto o dos.

- Ejercicio largo. Programar en la FPGA un sistema de generación de sinusoidales (digitales) de frecuencia variable (*Direct Digital Synthesis* o DDS), concepto explicado en la asignatura Circuitos Para Comunicaciones (impartida en el primer cuatrimestre).

D. Planteamiento de las prácticas de laboratorio

Se emplea la tarjeta DE-2 de Altera (véase Figura 3), con una FPGA *Cyclone-II*®. Dicha tarjeta incluye multitud de elementos apropiados para la realización de prácticas de carácter general: LEDs, *displays* de 7 segmentos, pantalla de cristal líquido, pulsadores, interruptores, etc. Incluye también elementos para el desarrollo de prácticas más complejas, como memorias SRAM, FLASH y SDRAM, lector de tarjetas SD, etc. Por último, incluye numerosos elementos muy orientados a la temática del máster, como son un CODEC de audio y los correspondientes conectores, salida VGA, Ethernet, conector DB-9 y circuito integrado MAX-232 para comunicación por puerto serie, conectores USB (*host* y *slave*), etc.

Las prácticas se han desarrollado tratando de maximizar el trabajo autónomo y minimizando el planteamiento de prácticas guiadas. Asimismo, tratan de vincularse al máximo con la temática del máster y el resto de asignaturas del mismo (véase Tabla 1). En relación al primer aspecto mencionado, cabe decir que se ha propuesto una posible mejora de cara a años futuros.

Únicamente las dos primeras prácticas son guiadas y se centran en que los alumnos aprendan el manejo del *software* específico para el desarrollo de los programas en VHDL (Quartus®) y en su simulación (ModelSim®). En estas prácticas el profesor les explica, paso a paso y mediante el desarrollo de un ejemplo, el uso de estos programas. Estas prácticas se realizan en las primeras semanas, por lo que los alumnos no han podido ver la programación básica mediante VHDL (instrucciones concurrentes y secuenciales). Por lo tanto, en los ejemplos se emplea programación gráfica

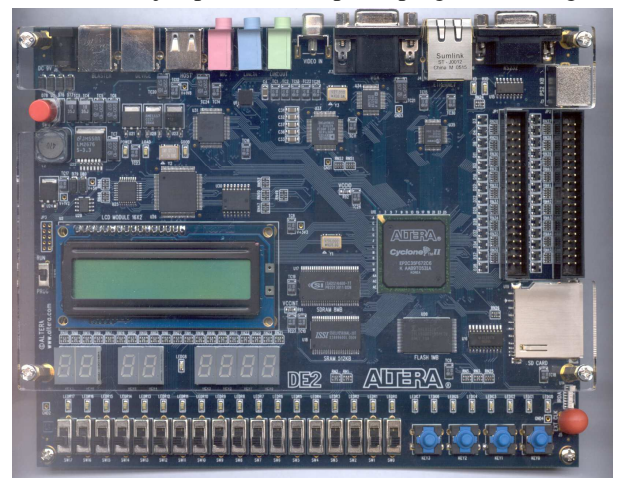


Figura 3. Fotografía de la placa de práctica DE-2

Tabla 1. Descripción, tareas obligatorias y voluntarias y peso de cada práctica en la nota final

Nº	Descripción	Tareas obligatorias	Tareas opcionales	% en la nota final
1	Desarrollar un sumador de 4 bits con acarreo, que muestre los dos sumandos y el resultado	Sumar dígitos decimales, los números deben parpadear con dos frecuencias distintas y debe ser posible apagarlos	No hay	0
2	Simular el proyecto anterior	Simulación con ModelSim de todos los casos posibles y comprobación de funcionamiento correcto	No hay	0
3	Juego Gato-Ratón	Ratón (LED encendido) que se desplace de un lado a otro y que rebote en las paredes levantadas (interruptores subidos).	Velocidad configurable, movimiento aleatorio, detección de ratón atrapado, conteo de veces atrapado, tiempo máximo de juego, número de jugadores, etc.	10
4	Cartel de mensajes	Almacenar 16 mensajes distintos de 8 dígitos de longitud. Dichos mensajes deben ser configurables al entrar en modo memorizar Elección de qué mensaje mostrar en los <i>displays</i> de 7 segmentos. El mensaje debe rotar.	Velocidad de rotación configurable, mensajes mayores de 8 dígitos, mapa de caracteres extendido, uso de la pantalla de cristal líquido, etc.	15
5	Configuración del CODEC mediante I2C	Protocolo de comunicación I2c que establezca al CODEC como esclavo y permita seleccionar un registro y cargar el valor deseado para configurar un cierto aspecto del CODEC.	Detección de errores de comunicación, selección de la velocidad de comunicación, etc.	5
6	Desarrollo de un piano digital	Introducción al CDA de las señales adecuadas para generar distintos tonos musicales.	Empleo de DDS para la generación de las señales, memorización de melodías, melodías pre-establecidas, etc.	20

mediante bloques. Con este planteamiento se cubren dos objetivos. Por un lado, los alumnos no tienen que esperar a haber finalizado los bloques relativos a la programación concurrente y secuencial para poder desarrollar programas. Por otro lado, se logra un ecualizado de los alumnos. Según su procedencia, hay alumnos que ya han programado FPGAs o CPLDs mediante Quartus. Otros, han programado usando Vivado o ISE (plataformas de Xilinx) y otros no han visto nada relativo a ninguna de ambas. Estas dos sesiones de prácticas sirven para nivelar a los alumnos, siendo un repaso para los primeros, una adaptación a una nueva plataforma para los segundos o un temario nuevo para los terceros. En cualquiera de los casos, al finalizar la segunda práctica todos tienen el nivel mínimo necesario para ser autónomos en el manejo de Quartus y ModelSim. No obstante, como se explica en el siguiente apartado, esto es un aspecto que se ha cambiado para el planteamiento del siguiente año, ya que es mejorable.

Las siguientes 4 prácticas (para un total de 6) siguen el mismo esquema: los alumnos deben trabajar sobre un proyecto determinado y entregarlo a las dos semanas. Dicho proyecto consta de una parte obligatoria y de partes opcionales. La parte obligatoria, si está correctamente desarrollada y es correctamente defendida por el alumno, supone obtener un 5 sobre 10 en dicha práctica (véase apartado de evaluación). Los trabajos adicionales, en función de su dificultad, número y exposición y defensa, complementan la nota hasta llegar al 10 sobre 10. Es importante que la práctica 3 comience cuando se ha finalizado la explicación del bloque relativo a las instrucciones secuenciales. De esta forma, los alumnos pueden realizar las prácticas (a excepción de las dos primeras, que emplean programación gráfica) pudiendo usar, como mínimo, todas las instrucciones básicas del lenguaje VHDL (concurrente y secuencial). Es cierto que aún restan por explicar otros bloques con aspectos más avanzados y

específicos del VHDL. Esos aspectos se pueden poner en juego de forma gradual en prácticas posteriores a la tercera.

Las prácticas tienen el siguiente objetivo:

- Práctica 3. Desarrollo de un juego en el que se debe capturar un ratón (representado por un LED encendido que se va “moviendo”) entre dos interruptores activados (véase la Figura 3, donde se puede ver que hay un LED encima de cada interruptor). Esta práctica se desarrolla una vez los alumnos han visto todas las instrucciones relativas a la programación concurrente y secuencial (bloques 3 y 4) y su finalidad es que los alumnos las pongan en prácticas, traten de desarrollar un código con ellas, encuentren problemas y los soluciones, etc. Su temática no es específica de la Ingeniería de Telecomunicaciones, pero sí es atractiva y logra captar la atención de los alumnos desde el primer momento [13].
- Práctica 4: desarrollo de un cartel luminoso capaz de almacenar y mostrar distintos mensajes. Los alumnos practican el uso de una memoria RAM, la interpretación de los cronogramas de tiempos mínimos y máximos a respetar, etc.
- Práctica 5: control de los registros de configuración del CODEC de audio mediante protocolo de comunicación I2C. El alumno pone en práctica el desarrollo de protocolos de comunicación y el de máquinas de estado.
- Práctica 6: desarrollo de un piano digital. Mediante el uso del CODEC deben ser capaces de generar tonos, reproducir melodías sencillas, etc. Esta práctica potencia el desarrollo de otro protocolo de comunicación distinto al anterior (I2S o DSP). Además, permite el empleo de otras técnicas propias

de la ingeniería de telecomunicación, tales como filtros FIR [14] o DDSs [15].

Los alumnos pueden hacer una práctica adicional voluntaria, centrada en el análisis de los datos digitales provenientes del conversor analógico-digital del CODEC. En concreto, el análisis de la frecuencia y amplitud de los tonos introducidos al mismo desde la entrada de micrófono. La repercusión de esta práctica adicional en la evaluación se describe en el siguiente apartado.

Por último, a principio de curso se da la opción a los alumnos de convalidar parte o todas las prácticas entregables por la realización de uno o varios proyectos de su elección. Al margen del incremento en la motivación que esto supone, hay que remarcar que en casos puntuales los alumnos ya conocen cuál será su trabajo fin de máster, pues es continuación de su proyecto fin de grado o ya está en contacto con alguna empresa. En esos casos, ya pueden intuir si les puede resultar interesante desarrollar un programa en VHDL. Hay que destacar que esta opción se plantea a principio de curso y debe ser aceptada por el profesor responsable tras un análisis del proyecto propuesto que evite grados de dificultad demasiado altos o bajos.

E. Relación con el resto de asignaturas del máster

Se debe tener claro que dentro del máster, los contenidos de esta asignatura constituyen un medio y no un fin en sí mismos. Es decir, con esta asignatura los alumnos deben ser capaces de desarrollar las técnicas y procedimientos que se le explican en otras materias. Por lo tanto, en la medida de lo posible, los ejercicios (especialmente los más largos), deben tratar de girar en torno a la temática de otras asignaturas. Para lograrlo, cada año se llevan a cabo dos acciones orientadas a ello:

- En la encuesta que se realiza a los alumnos a final de curso se les pregunta por aspectos de otras asignaturas que creen o les gustaría haber trabajado desde la perspectiva del VHDL. Esta forma de proceder ha resultado especialmente útil dado que los alumnos son aquellos sobre los que convergen todas las asignaturas y tienen, por tanto, el punto de vista más general.
- Se ha preguntado a otros profesores sobre qué aspectos de sus asignaturas son más proclives a estar programados en un sistema digital (sea una FPGA, un DSP o un ordenador). Aunque también ha dado buenos resultados, debe tenerse en cuenta que un profesor de una cierta asignatura de telecomunicaciones no tiene por qué ser capaz de decidir cuáles de los sistemas que plantea en su asignatura son buenos candidatos a ser programados específicamente en una FPGA.

F. Evaluación del alumnado

La evaluación tiene un planteamiento sencillo, 50% el examen final y 50% las prácticas (véase Tabla 1), en las que las dos primeras, al ser de nivelación, no tienen influencia en la nota final. Comentarios adicionales a este respecto pueden verse en el siguiente apartado. Por otro lado, la práctica adicional voluntaria supone un aumento de la nota final de hasta un punto y la posibilidad de obtener Matrícula de Honor.

IV. CONCLUSIONES Y MEJORAS

Al finalizar cada curso, se realiza una encuesta a los alumnos para que puedan evaluar diversos aspectos de la asignatura, proponer mejoras, realizar críticas, etc. Dado que este es el segundo año de andadura del máster, no tiene sentido un análisis estadístico o cuantitativo de los resultados obtenidos en la evaluación de diversos apartados (calidad de los apuntes, satisfacción con el profesor y la asignatura, etc.). Baste decir que la nota media de la asignatura y los profesores que lo imparten es superior a 9. Sí resulta de especial interés comentar aquellas críticas, comentarios y sugerencias que han permitido realizar cambios, detectar nichos de mejora o, sencillamente, detectar problemas a corregir (aspectos con nota inferior a 9). Al margen de la encuesta, existen una serie de mejoras que han sido detectadas por los profesores involucrados en la asignatura y que también se recogen en este apartado.

A. El peso del examen no es demasiado alto en relación a las prácticas

Los alumnos han evaluado como excesivo el peso del examen en relación a las prácticas. No obstante, un análisis pormenorizado permite deducir que no es tan grave como primeramente puede parecer. La encuesta se realiza justo cuando se ha finalizado el periodo lectivo y antes de realizar los exámenes. Por lo tanto, los alumnos consideran que todo el trabajo realizado durante las prácticas puede perderse si el examen es demasiado difícil, se tiene un mal día, etc. Es decir, dicho juicio se emite con cierto grado de subjetividad. Una vez realizado el examen, se ha preguntado a los alumnos si la dificultad del examen y las preguntas que en él se incluían eran acordes al peso del mismo, justas, y bien coordinadas con las prácticas. En todos los casos, los alumnos admitieron que se trataba de un examen justo, asequible y que cumplía su finalidad. A este último respecto, debe decirse que la finalidad del examen, y así se hace saber a los alumnos, es doble. Por un lado, evaluar ciertos aspectos que no suelen emplearse en las prácticas (uso de funciones, funciones de resolución, etc.) y, por otro, evitar que los alumnos copien las prácticas. En el caso del primer punto, debe decirse que se deja a los alumnos absoluta libertad en la forma de desarrollar los códigos de las prácticas, pero se les recomienda que traten de aprovecharlas y emplear cuántos más aspectos explicados en teoría mejor. No obstante, la experiencia indica que suelen emplear, en general, un rango limitado de todos los recursos que ofrece el VHDL. Al incluir la evaluación de dichos aspectos en el examen, y ser los alumnos conscientes de ello, tienden a estudiar todos los temas y usar los conocimientos adquiridos durante el desarrollo de los programas. En el caso del segundo punto, la entrega de las prácticas implica una defensa frente al profesor, que hace preguntas sobre la misma, propone pequeños cambios para comprobar que el alumno conoce el código, etc. Es decir, la propia evaluación de las prácticas constituye un método que permite detectar a aquellos alumnos que han copiado o no han trabajado en sus prácticas. La realización del examen es una medida adicional.

B. La cantidad de clases expositivas es excesiva en relación a las prácticas de laboratorio

Se trata de un problema detectado por los propios profesores desde el comienzo del primer curso. La solución es tan sencilla como emplear horas de clases expositivas para la realización de prácticas de laboratorio. No obstante, se debe tener en cuenta que esto implica que la carga docente de los profesores aumenta sin que ello se vea reflejado en su plan docente. En concreto, de las 20 horas de clases expositivas (8 fueron convertidas en horas de prácticas de aula), 3-4 se destinan a prácticas de laboratorio.

C. Los ejercicios sencillos deberían limitarse a los primeros temas.

Durante ambos años, todos los bloques y temas han tenido ejercicios cortos (véase III.C) cuya resolución llevaban a cabo los alumnos durante la clase y para lo que tenían unos 5-20 minutos. Los comentarios de los alumnos apuntan a que eso sólo es útil durante los bloques relativos a las instrucciones concurrentes y secuenciales. Una vez pasados esos bloques, los alumnos prefieren omitir, o limitar, dicho tipo de ejercicios. De hecho, tiene bastante sentido ya que dichos bloques resultan ser los más complicados para los alumnos dado el cambio de paradigma en la programación (pasan de programar *software* a programar *hardware*). Por lo tanto, el año que viene se procederá a modificar el planteamiento de los ejercicios cortos de los bloques relativos a los aspectos avanzados y a la programación jerárquica. Esto no quiere decir que se eliminen, sino que salvo en casos puntuales, se procederá a su resolución como si de ejemplos se trataran (planteando una o dos preguntas específicas, pero explicando la solución de forma inmediata).

D. El primer bloque debería acortarse

Para tratar de mantener relación con la asignatura de diseño microelectrónico, en el primer bloque se incluía una descripción pormenorizada de todas las tecnologías existentes para la implementación de sistemas digitales (lógica discreta, ASICs, *full custom*, etc.). Aun sin haber una asignatura con la que tratar de mantener una vinculación, esto suele ser habitual en asignaturas centradas en el lenguaje VHDL como un intento de contextualizar a las FPGAs. Se ha optado por resumir dicho tema y centrarlo únicamente en la estructura interna de las FPGAs. El resto del tema se deja como lectura obligada para los alumnos, ya que por otro lado, su sencillez así lo permite.

E. Propuesta de temas y ejercicios nuevos.

Como ya se ha comentado, para tratar de relacionar esta asignatura con el resto de asignaturas del máster, se ha entablado contacto con el resto de profesores del máster. No obstante, debe tenerse en cuenta que eso tiene una eficiencia limitada, ya que cada profesor es experto en las materias que imparte, pero no tiene por qué serlo en VHDL. Por lo tanto, no siempre es capaz de identificar nichos donde el VHDL y el contenido de sus asignaturas pueden tener relación. Por ello, incluir en la encuesta a los alumnos propuesta de ideas nuevas, aspectos de otras asignaturas que encajen dentro de la programación VHDL ha tenido buenos resultados durante los dos años de funcionamiento de la asignatura.

F. Necesidad de tener placas y estar en el laboratorio para hacer las prácticas

El desarrollo de las prácticas implica que éstas deben funcionar en las placas de prácticas. Eso, a su vez, hace que los alumnos tiendan a trabajar probando sus diseños directamente en ellas sin emplear los simuladores disponibles en ningún momento. Esto, unido a que el laboratorio no siempre está disponible ya que alberga la realización de prácticas de otras asignaturas, hace que los alumnos hayan pedido un mayor número de placas de prácticas y un lugar alternativo donde trabajar. Analizando la situación, el problema no está en el número de prácticas, sino en la metodología que siguen los alumnos, que depuran sus programas directamente probándolos sobre la placa. La solución pasa por insistir en que el primer paso de la depuración de un programa debe ser el empleo de simuladores, donde es posible conocer la evolución de variables y señales internas y realizar una simulación paso a paso (con la dificultad añadida de tratarse de instrucciones de ejecución concurrente). Esta insistencia ha tenido resultados positivos este año, donde un mayor número de alumnos ha empleado de forma constante los simuladores. Por otro lado, el próximo año se va a pedir que cada práctica vaya acompañada de una simulación que muestre el correcto funcionamiento del programa, como una medida adicional para potenciar y fomentar el uso de simuladores.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Economía a través del proyecto DPI2013-47176-C2-2-R y MINECO-15-DPI2014-56358-JIN, y el Gobierno del Principado de Asturias a través del proyecto FC-15-GRUPIN14-143 y los fondos FEDER.

REFERENCIAS

- [1] Miaja, P.F.; Lamar, D.G.; de Azpeitia, M.; Rodríguez, A.; Rodríguez, M.; Hernando, M.M., "A Switching-Mode Power Supply Design Tool to Improve Learning in a Power Electronics Course," in *Education, IEEE Transactions on*, vol.54, no.1, pp.104-113, Feb. 2011
- [2] Stolk, J.; Martello, R.; Geddes, J., "Work in progress - building autonomous students: Modeling curricular approaches for lifelong learning," in *Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE '07. 37th Annual*, vol., no., pp.F3B-20-F3B-21, 10-13 Oct. 2007
- [3] Aguilar-Pena, J.D.; Muñoz-Rodríguez, F.J.; Rus-Casas, C.; Romero-Moreno, M.; Pérez-Higueras, P.J., "Educational tools in order to promote the self-learning. Practical case of study: Dimex SFCR," in *Tecnologías Aplicadas a la Enseñanza de la Electrónica (Technologies Applied to Electronics Teaching) (TAE), 2014 XI*, vol., no., pp.1-6, 11-13 June 2014
- [4] Carrillo, J.M.; Pérez-Aloe, R.; Ausin, J.L.; Duque, J.F.; Carmona, P., "Coordination of electronics courses of the Master Degree in Industrial Engineering by means of a quality assurance system," in *Design of Circuits and Integrated Systems (DCIS), 2015 Conference on*, vol., no., pp.1-4, 25-27 Nov. 2015
- [5] Tovar, E.; Castro, M.; Llamas, M.; Arcega, F.; Jurado, F.; Mur, F.; Sanchez, J.A.; Plaza, I.; Falcone, F.; Dominguez, M., "Work in Progress: Modeling the best practices of adaptation to European credit transfer system in technical teachings as improvement to the mobility of students," in *Frontiers in Education Conference, 36th Annual*, vol., no., pp.1-2, 27-31 Oct. 2006
- [6] Arias, M.; Pando, P.; Rodríguez, A.; Miaja, P.F.; Vazquez, A.; Fernandez, M.; Lamar, D.G., "The Master's Thesis: An Opportunity for

- Fostering Presentation Skills," in *Education, IEEE Transactions on* , vol.57, no.1, pp.61-68, Feb. 2014
- [7] Kulturel-Konak, S.; Konak, A.; Esparragoza, I.E.; Kremer, G.E.O., "Assessing professional skills in STEM disciplines," in *Integrated STEM Education Conference (ISEC), 2013 IEEE* , vol., no., pp.1-4, 9-9 March 2013
- [8] Vaid, V., "Comparison of different attributes in modeling a FSM based vending machine in 2 different styles," in *Embedded Systems (ICES), 2014 International Conference on* , vol., no., pp.18-21, 3-5 July 2014
- [9] dos Santos, S.C.; Furtado, F.; Lins, W., "xPBL: A methodology for managing PBL when teaching computing," in *Frontiers in Education Conference (FIE), 2014 IEEE* , vol., no., pp.1-8, 22-25 Oct. 2014
- [10] Arias, M.; Rodriguez, A.; Lamar, D.G.; Linera, F.; Hernando, M., "Influence of PBL practical classes on microcontroller-based digital systems learning," in *Education Engineering (EDUCON), 2010 IEEE* , vol., no., pp.1777-1782, 14-16 April 2010
- [11] Montanana, J.M.; Lemus-Zuniga, L.G.; Alanis Garza, A.; Benlloch-Dualde, J.V., "Teaching strategy on VHDL course based on participative learning," in *Information Technology Based Higher Education and Training (ITHET), 2015 International Conference on* , vol., no., pp.1-6, 11-13 June 2015
- [12] Lamar, D.G.; Miaja, P.F.; Arias, M.; Rodriguez, A.; Rodriguez, M.; Sebastian, J., "A project-based learning approach to teaching power electronics: Difficulties in the application of Project-Based Learning in a subject of Switching-Mode Power Supplies," in *Education Engineering (EDUCON), 2010 IEEE* , vol., no., pp.717-722, 14-16 April 2010
- [13] Yulia; Adipranata, R., "Teaching object oriented programming course using cooperative learning method based on game design and visual object oriented environment," in *Education Technology and Computer (ICETC), 2010 2nd International Conference on* , vol.2, no., pp.V2-355-V2-359, 22-24 June 2010
- [14] Song Yu; Feng Yisi, "16-order fir filter design based on MATLAB and its Quartus II simulation," in *Computer, Mechatronics, Control and Electronic Engineering (CMCE), 2010 International Conference on* , vol.5, no., pp.44-46, 24-26 Aug. 2010
- [15] Fang Yi-yuan; Chen Xue-jun, "Design and simulation of DDS based on Quartus II," in *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on* , vol.2, no., pp.357-360, 10-12 June 2011